

The End of Moore's Law?

The current economic boom is likely due to increase in computing speed and decreases in price. Now there are some reasons to think that the party may be ending.

By Charles C. Mann

From today's perspective, it seems clear that Gordon Moore got lucky. Back in 1965, *Electronics* magazine asked Moore—then research director of electronics pioneer Fairchild Semiconductor—to predict the future of the microchip industry. At the time, the industry was in its infancy; Intel, now the world's biggest chip-maker, would not be founded (by Moore, among others) for another three years. Because few chips had been manufactured and sold, Moore had little data to go on. Nonetheless, he confidently argued that engineers would be able to cram an ever-increasing number of electronic devices onto microchips. Indeed, he guessed that the number would roughly double every year—an exponential increase that has come to be known as Moore's Law.

At first, few paid attention to Moore's prediction. Moore himself admitted that he didn't place much stock in it—he had been "just trying to get across the idea [that] this was a technology that had a future. But events proved him right. In 1965, when Moore wrote his article, the world's most complex chip was right in his lab at Fairchild: It had 64 transistors. Intel's new-model Pentium III, introduced last October, contains 28 million transistors. The sustained explosion of microchip complexity—doubling year after year, decade after decade," Lillian Hoddeson and Michael Riordan write in *Crystal Fire*, their history of the transistor, "has no convenient parallel or analogue in normal human experience."

The effect of Moore's Law on daily life is obvious. It is why today's \$3,000 personal computer will cost \$1,500 next year and be obsolete the year after. It is why the children who grew up playing Pong in game arcades have children who grow up playing Quake on the Internet. It is why the word-processing program that fit on two floppy disks a decade ago now fills up half a CD-ROM—in fact, it explains why floppy disks themselves have almost been replaced by CD-ROMs, CD-Rs and CD-RWs.

But these examples, as striking as they are, may understate the importance of Moore's Law. The United States is experiencing the longest economic boom since the 1850s, when the federal government first began collecting economic statistics systematically. The current blend of steady growth and low inflation is so unusually favorable that many economists believe the nation is undergoing fundamental change. And the single most important factor driving the change, these economists say, is the relentless rise in chip power. "What's sometimes called the 'Clinton economic boom,'" says Robert Gordon, an economist at Northwestern University, "is largely a reflection of Moore's Law." In fact, he says, "the recent acceleration in productivity is at least half due to the improvements in computer productivity."

If Gordon is right, it is unfortunate that just as economists are beginning to grasp the importance of Moore's Law, engineers are beginning to say that it is in danger of petering out.

The age of digital electronics is usually said to have begun in 1947, when a research team at Bell Laboratories designed the first transistor. But Moore's Law, the driving force of the digital era, is pegged to another, lesser-known landmark: the invention of the integrated circuit. John Bardeen, Walter Brattain and William Shockley won a Nobel Prize for the transistor. Jack Kilby, the Texas Instruments engineer who came up with the integrated circuit, didn't win anything. But in many ways it was his creation, not the transistor, that most shook the world.

In May 1958 Kilby was hired by Texas Instruments, the company that pioneered the silicon transistor. The company had a mass vacation policy; almost everyone was thrown out of the office for the first few weeks in July. Being newly hired, Kilby had no time off. He found himself almost alone in the deserted plant.

Transistors, diodes, capacitors and other now-familiar electronic devices had just been invented, but already some far-sighted people, many of them in the Pentagon, were thinking about lashing together these individual components into more complex circuits. Texas Instruments was trying to hook up with the Army's Micro-Module program, in which individual components were built on small wafers and stacked like so many poker chips. Kilby thought this approach was ludicrous—a kludge, in engineer's slang. By the time a module was large enough to do something interesting, the stack of wafers would be ridiculously big and cumbersome.

On July 24—exactly a month after Bell Labs celebrated the 10th anniversary of the public unveiling of the transistor—inspiration paid a visit to Kilby in the empty factory. Instead of wiring together components in modules, he wrote in his lab notebook, engineers should scatter "resistors, capacitors and transistors & diodes on a single slice of silicon." Classic inventors' stories usually include a chapter about how management ignores the inventor's brilliant new idea. At Texas Instruments, Kilby's boss immediately asked him to build a prototype. By September, Kilby had assembled one. It was simple and crude, but it worked. The company filed for a patent on its revolutionary "Solid Circuit" in February 1959.

Two weeks before the filing, a similar idea occurred to Robert Noyce, an engineer at Fairchild Semiconductor, one of the first startup tech firms in Silicon Valley. (Noyce and Moore would later leave Fairchild to found Intel.) Whereas Kilby had linked the components of his integrated circuit by gold wires and solder, Noyce realized the connections could be painted on the silicon with a kind of stencil—a photomicrolithograph, to be precise. Noyce's bosses, like Kilby's, were enthusiastic. And in July Fairchild, too, filed for a patent.

Litigation inevitably ensued. It lasted 10 years and ended with the companies fighting to a draw. But while the lawyers argued, both companies raced to create ever-more-sophisticated integrated circuits—"chips," as they came to be called. The first chip appeared on the market in 1961, to less than universal acclaim; engineers, accustomed to designing their own circuits, initially regarded these prefabricated gizmos as annoyances. But the companies kept going. By 1964 some chips had as many as 32 transistors; when Moore wrote his article in 1965, a chip in his R&D lab had twice as many.

One component (1959), 32 (1964), 64 (1965)—Moore put these numbers on a graph and connected the dots with a line. "The complexity [of cheap integrated circuits] has increased at a rate of roughly a factor of two per year," he wrote. Then he got

out a ruler and extended the line into the future. It sailed off the top of his graph and into the stratosphere. "Over the longer term...", Moore argued, "there is no reason to believe [the rate of increase] will not remain constant for at least 10 years." In other words, the companies that were then laboring to create microchips with 64 components would in a decade be manufacturing microchips with over 65,000 components—a jump of more than three orders of magnitude.

Moore's Law was not, of course, a law of nature. It was more like an engineer's rule of thumb, capturing the pattern Moore had discerned in the early data on microchip production. But law or no, by 1975 engineers were designing and manufacturing chips a thousand times more complex than had been possible just 10 years before—just as Moore had predicted. That year, Moore revisited his prediction at the annual International Electron Devices Meeting of the Institute of Electrical and Electronics Engineers, the professional association of electrical engineers. Acknowledging the increasing difficulty of the chip-making process, Moore slightly revised his "law." From that point on, he said, the number of devices on a chip would double every two years. This prediction proved correct, too. Today, some people split the difference and say that microchip complexity will double every 18 months; other people loosely apply the term "Moore's Law" to any rapidly improving aspect of computing, such as memory storage or bandwidth.

Despite the fuzziness about exactly what Moore's Law states, its gist is indisputable: Computer prices have fallen even as computer capabilities have risen. At first glance, this is unsurprising. Although digital gurus often herald the advent of better products at lower costs as an unprecedented boon, it is in fact an economic commonplace. A car from 1906, which by today's standards is barely functional, then cost the equivalent of \$52,640, according to a study by Daniel Raff of the Wharton School of Business and Manuel Trajtenberg of Tel Aviv University. Nonetheless, the digital gurus have a point. The improvements in computer chips have been unprecedentedly rapid—"manna from heaven," in the phrase of Erik Brynjolfson, an economist at MIT's Sloan School of Management. "It's this lucky combination of geometry and physics and engineering," he says. "The technical innovation is normal, but the rate at which it is occurring is highly unusual."

Drawn by rapidly improving products at rapidly falling prices, U.S. spending on computers has risen for the last twenty years at an average annual clip of 24 percent—a Moore's Law of its own. In 1999, U.S. companies spent \$220 billion on computer hardware and peripherals, more than they invested in factories, vehicles or any other kind of durable equipment. Computers became so ubiquitous and powerful that it became commonplace to hear the claim that the nation was in the middle of a "digital revolution." Moore's Law, the pundits claim, has created a "new economy."

Maybe so, but for a number of years the evidence didn't seem to be there. Like everyone else, economists had been discovering the wonders of the inexpensive beige boxes now on their desks. They kept waiting to see the rewards of computing pop up in the government statistics on income, profits and productivity. But it didn't happen. Throughout the 1980s and the first part of the 1990s the huge national investment in digital technology seemed to have almost no payoff; Moore's Law ended up boosting profits for chip-makers, but hardly anyone else. "We see the computer age everywhere except in the productivity statistics," the Nobel Prize-winning MIT economist Robert M. Solow remarked in 1987.

The puzzle—huge expenditures with little apparent benefit—became known as the "productivity paradox." Not only were these new technical wonders not useful, some researchers argued, they might actually be harmful. Since 1980 the service industries alone have spent more than a trillion dollars on computer hardware and software. Yet Stephen S. Roach, chief economist of Morgan Stanley, suggested in 1991 that this had merely transformed the service sector from an industry characterized by variable labor costs to one that was increasingly dominated by fixed hardware costs. The least productive "portion of the economy," Roach argued, "[is] the most heavily endowed with high-tech capital"—the more computers, in other words, the less value.

"Look at hotel checkouts," says Lester Thurow, one of Brynjolfson's colleagues at Sloan. "They're completely computerized now, but nobody seems to be doing anything faster. The same thing at the supermarket—you wait in line just as long as you used to wait." To Thurow, the service sector, which is almost three-quarters of the economy, "seems at first glance to have swallowed vast amounts of computing power without a trace."

"Nobody could understand it," says Hal Varian, an economist at the School of Information Management Systems at the University of California, Berkeley. "On the face of it, the statistics coming out of the government were saying that this massive investment was senseless. In the past, technological innovation has almost invariably increased living standards—look at electricity, railroads, telephones, antibiotics. And here was Moore's Law—innovation of unprecedented rapidity—that seemed to create nothing for human welfare. But if computers had so little payoff, why was everyone rushing to buy the damned things?"

To people like Varian, what happened at the Federal Trade Commission is an example of what should have been going on all over the country. In the mid-1980s, the FTC gave a personal computer to every staffer in the Bureau of Economics, its in-house economic advisory board. "The computers had two effects," recalls a former FTC economist. "For the first three months, the economists spent long hours worrying about their fonts"—that is, about making their letters and memos look pretty. "Six months later, they got rid of the steno pool."

For economists, this is a textbook example of increased productivity. The agency produced the same number of reports with fewer people, which means the per-capita production of economics was higher. (More precisely, this is an example of increased labor productivity; economists also use another, more complex measure, multifactor productivity, but for most purposes the two can be treated together.)

Spread throughout the economy, higher productivity means higher wages, higher profits, lower prices. Productivity increases aren't necessarily painless, as the dismissed stenographers at the FTC found out. But history shows that workers displaced by productivity-enhancing technology usually find other, better jobs. In the long run raising productivity is essential to increasing the national standard of living. "In some sense," Thurow says, "if you could only know one number about an economy, you'd like to know the level and the rate of growth of productivity, because it underlies everything else."

After World War II, the United States spent decades with productivity growing at an average rate of almost 3 percent a year—enough, roughly speaking, to double living

standards every generation. In 1973, however, productivity growth suddenly slowed to 1.1 percent, far below its previous level. Nobody knows why. "The post-1973 productivity slowdown," says Jack Triplett of the Brookings Institute, "is a puzzle that has so far resisted all attempts at solution."

The effects of the slowdown, alas, are well-known. At that slower rate, living standards double in three generations, not one. The result was stagnation. Wage-earners still won raises, but employers, unable to absorb the extra costs with higher productivity, simply passed the increase into higher prices, which canceled the benefit of higher wages. Unsurprisingly, economists say, the unproductive 1970s and 1980s were years of inflation, recession, unemployment, social conflict and enormous budget deficits.

In 1995, productivity changed direction again. Without any fanfare, it abruptly began rising at an annual average clip of almost 2.2 percent—a great improvement from the 1980s, though still less than the 1960s. At first, most researchers regarded the increase as a temporary blip. But gradually many became convinced that it was long-lasting. "It was certainly something we discussed a lot at [Federal Reserve Board] meetings," says Alice Rivlin, a Brookings economist who recently left the board. "You know, 'Is this increase real?' By now, I think most economists believe it is." The implications, in her view, are enormous: Renewed productivity growth means that more people are more likely to achieve their dreams.

Although Rivlin is co-leading a Brookings study to determine the cause of the new productivity boom, she and many other economists believe it is probably due to computerization. "Moore's Law," she says, laughing, "may finally be paying off."

There are two reasons for this belief, says Alan S. Blinder, a Princeton University economist. First, the acceleration in productivity happened "co-terminously" with a sudden, additional drop in computer costs. Second, the coincidence that productivity rose just as business adopted the Internet "is just too great to ignore."

In the mid-'90s, Blinder says, "the rate of computer deflation moved from minus 10 percent to minus 25 percent per annum. And although the computer industry is a small fraction of GNP—less than 2 percent—the drop in costs has been so severe that as a matter of arithmetic it knocks a noticeable piece off the overall price index." In fact, the recent declines in the price of computers are so big that Gordon, the economist at Northwestern, argues that they largely explain the bump in productivity—except for durable goods manufacturing the economy is stagnant.

Gordon's argument is "too extreme," in the view of Chris Varvares, president of Macroeconomic Advisers, an economic modeling firm in St. Louis. "Why would business invest in all this equipment if they didn't have the expectation of a return? And since it's gone on for so long, why wouldn't they have the reality?" Instead, he says, computers and the Internet are finally paying off in ways that statistics can measure. When banks introduce automated teller machines, the benefits don't show up in government statistics. Bank customers are better off, because they can withdraw and deposit money at any time and in many more places. But the bank itself is still doing what it did before. "The benefits are captured by consumers, and don't show up in the bottom line as output," says Varvares. Only recently, he argues, did computers hit a kind of critical mass; workers had so much digital power on their desks that it muscled its way into the statistics.

Not every economist agrees. "You'd like to be able to tell yourself a story about how something could be true," Thurow says. "In this case, are we saying that people suddenly figured out how to use computers in 1996?" No, other economists say, but businesses do need time to accommodate new technologies. Electricity took more than two decades to exert an impact on productivity, according to Stanford University economic historian Paul A. David. Computers simply encountered the same lag. But by now, Brynjolfson says, "computers are the most important single technology for improving living standards. As long as Moore's Law continues, we should keep getting better off. It will make our children's lives better."

The explosion in computer power has become so important to the future, these economists say, that everyone should be worried by the recent reports that Moore's Law might come to a crashing halt.

The end of Moore's Law has been predicted so many times that rumors of its demise have become an industry joke. The current alarms, though, may be different. Squeezing more and more devices onto a chip means fabricating features that are smaller and smaller. The industry's newest chips have "pitches" as small as 180 nanometers (billionths of a meter). To accommodate Moore's Law, according to the biennial "road map" prepared last year for the Semiconductor Industry Association, the pitches need to shrink to 150 nanometers by 2001 and to 100 nanometers by 2005. Alas, the road map admitted, to get there the industry will have to beat fundamental problems to which there are "no known solutions." If solutions are not discovered quickly, Paul A. Packan, a respected researcher at Intel, argued last September in the journal *Science*, Moore's Law will "be in serious danger."

Packan identified three main challenges. The first involved the use of "dopants," impurities that are mixed into silicon to increase its ability to hold areas of localized electric charge. Although transistors can shrink in size, the smaller devices still need to maintain the same charge. To do that, the silicon has to have a higher concentration of dopant atoms. Unfortunately, above a certain limit the dopant atoms begin to clump together, forming clusters that are not electrically active. "You can't increase the concentration of dopant," Packan says, "because all the extras just go into the clusters." Today's chips, in his view, are very close to the maximum.

Second, the "gates" that control the flow of electrons in chips have become so small that they are prey to odd, undesirable quantum effects. Physicists have known since the 1920s that electrons can "tunnel" through extremely small barriers, magically popping up on the other side. Chip gates are now smaller than two nanometers—small enough to let electrons tunnel through them even when they are shut. Because gates are supposed to block electrons, quantum mechanics could render smaller silicon devices useless. As Packan says, "Quantum mechanics isn't like an ordinary manufacturing difficulty—we're running into a roadblock at the most fundamental level."

Semiconductor manufacturers are also running afoul of basic statistics. Chip-makers mix small amounts of dopant into silicon in a manner analogous to the way paint-makers mix a few drops of beige into white paint to create a creamy off-white. When homeowners paint walls, the color seems even. But if they could examine a tiny patch of the wall, they would see slight variations in color caused by statistical fluctuations in the concentration of beige pigment. When microchip components were bigger, the similar fluctuations in the concentration of dopant had little effect. But

now transistors are so small they can end up in dopant-rich or dopant-poor areas, affecting their behavior. Here, too, Packan says, engineers have "no known solutions."

Ultimately, Packan believes, engineering and processing solutions can be found to save the day. But Moore's Law will still have to face what may be its most daunting challenge—Moore's Second Law. In 1995, Moore reviewed microchip progress at a conference of the International Society for Optical Engineering. Although he, like Packan, saw "increasingly difficult" technical roadblocks to staying on the path predicted by his law, he was most worried about something else: the increasing cost of manufacturing chips.

When Intel was founded in 1968, Moore recalled, the necessary equipment cost roughly \$12,000. Today it is about \$12 million—but it still "tends not to process any more wafers per hour than [it] did in 1968." To produce chips, Intel must now spend billions of dollars on building each of its manufacturing facilities, and the expense will keep going up as chips continue to get more complex. "Capital costs are rising far faster than revenue," Moore noted. In his opinion, "the rate of technological progress is going to be controlled [by] financial realities." Some technical innovations, that is, may not be economically feasible, no matter how desirable they are.

Promptly dubbed "Moore's Second Law," this recognition would be painfully familiar to anyone associated with supersonic planes, mag-lev trains, high-speed mass transit, large-scale particle accelerators and the host of other technological marvels that were strangled by high costs. If applied to Moore's Law, the prospect is dismaying. In the last 100 years, engineers and scientists have repeatedly shown how human ingenuity can make an end run around the difficulties posed by the laws of nature. But they have been much less successful in cheating the laws of economics. (The impossible is easy; it's the unfeasible that poses the problem.) If Moore's Law becomes too expensive to sustain, Moore said, no easy remedy is in sight.

Actually, that's not all that he said. Moore also argued that the only industry "remotely comparable" in its rate of growth to the microchip industry is the printing industry. Individual characters once were carved painstakingly out of stone; now they're whooshed out by the billions at next to no cost. Printing, Moore pointed out, utterly transformed society, creating and solving problems in arenas that Gutenberg could never have imagined. Driven by Moore's Law, he suggested, information technology may have an equally enormous impact. If that were the case, the ultimate solution to the limits of Moore's Law may come from the very explosion of computer power predicted by Moore's Law itself—"from the swirl of new knowledge, methods and processes created by computers of this and future generations."

The idea sounds far-fetched. But then Moore's Law itself sounded far-fetched in 1965.

For 35 years, computer chip speed has improved at a dependably exponential pace, fueling the productivity that underlies the economic boom. Is the party almost over? Join Charles Mann and TR readers in an [online forum](#) to discuss the implications.